



University of Pennsylvania
ScholarlyCommons

Technical Reports (CIS)

Department of Computer & Information Science

September 1992

Categorical Grammar

Mark Steedman
University of Pennsylvania

Follow this and additional works at: https://repository.upenn.edu/cis_reports

Recommended Citation

Mark Steedman, "Categorical Grammar", . September 1992.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-92-52.

This paper is posted at ScholarlyCommons. https://repository.upenn.edu/cis_reports/466
For more information, please contact repository@pobox.upenn.edu.

Categorial Grammar

Abstract

The paper is a review article comparing a number of approaches to natural language syntax and semantics that have been developed using categorial frameworks.

It distinguishes two related but distinct varieties of categorial theory, one related to Natural Deduction systems and the axiomatic calculi of Lambek, and another which involves more specialized combinatory operations.

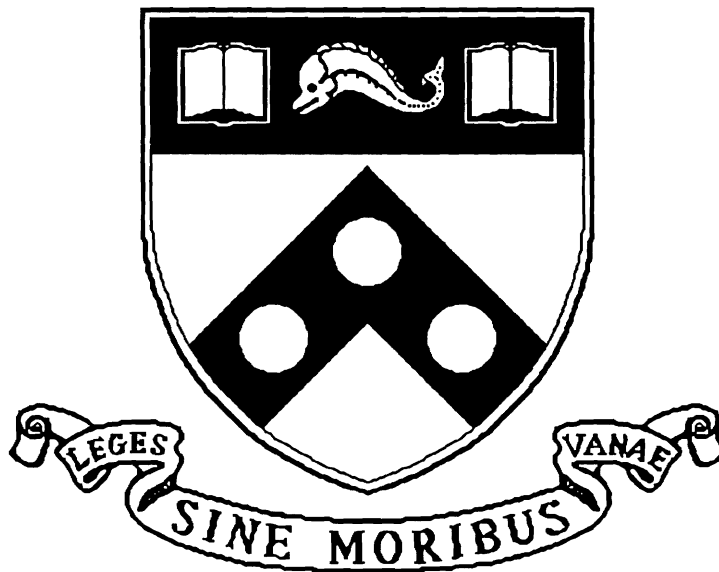
Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-92-52.

Categorical Grammar

MS-CIS-92-52
LINC LAB 230

Mark Steedman



University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

September 1992

CATEGORIAL GRAMMAR*

MARK STEEDMAN
University of Pennsylvania

September 2, 1992

Categorical Grammar (CG) is a term which covers a number of related formalisms that have been proposed for the syntax and semantics of natural languages and logical and mathematical languages. All are generalisations of a core context-free grammar formalism first explicitly defined by Ajdukiewicz 1935, but with earlier antecedents in the work of Husserl, Leśnewski, Frege, Carnap and Tarski on semantic and syntactic categories, ultimately stemming from work in the theory of types, (a tradition to which some recent work in CG shows signs of returning). The distinguishing characteristics of these theories are: an extreme form of *lexicalism* where the main and even entire burden of syntax is borne by the lexicon; the characterisation of constituents, both syntactically and semantically, as *functions* and/or *arguments*; the characterisation of the relation between syntax and semantics as *compositional*, with syntactic and semantic types standing in the closest possible relation, the former merely encoding the latter; a tendency to *freer surface constituency* than traditional grammar, the previously mentioned characteristic guaranteeing that all the non-standard constituents that CG sanctions are fully interpreted semantically.

Such grammars have been implicated in much work at the foundation of modern theories of natural language semantics. Like their theoretical cousins Tree Adjunction Grammars (TAG, Joshi et al 1987, Lexical Func-

*Thanks to Dick Oehrle for reading the draft and for many helpful comments. The work was supported in part by NSF grant nos IRI90-18513 and CISE IIP, CDA 88-22719, DARPA grant no. N00014-90-J-1863, and ARO grant no. DAAL03-89-C0031.

tional Grammar (LFG, Bresnan 1982), and Generalised Phrase Structure Grammar (GPSG, Gazdat et al 1985), they have also recently provided an important source of constrained alternatives to transformational rules and their modern derivatives for formal theories of natural language syntax. In the syntactic arena, categorial grammars have been claimed to have significant advantages as explanatory and unifying theories of unbounded constructions, including coordination and relative clause formations, of constructions that have been held to involve “reanalysis”, of phonological phrasing associated with intonation, of numerous clause-bounded phenomena including reflexive binding, raising, and control, and of analogous discontinuous phenomena in morphology.

§1 PURE CATEGORIAL GRAMMAR

In a categorial grammar, all grammatical constituents, and in particular all lexical items, are associated with a type or “category” which defines their potential for combination with other constituents to yield compound constituents. The category is either one of a small number of “basic” categories, such as *NP*, or a “functor” category. The latter have a type which identifies them as functions mapping arguments of some type onto results of some (possibly different) type. For example, English intransitive verbs like *walks* are most naturally defined as functions from nounphrases *NP* on their left to sentences *S*. English transitive verbs like *sees* are similarly defined as functions from nounphrases *NP* on their right to the aforementioned intransitive verb category. Apart from a language-particular specification of directionality, such categories merely reflect the types of the semantic interpretations of these words.

There are several different notations for directional categories. The most widely used are the “slash” notations variously pioneered by Bar-Hillel 1953, Lambek 1958, and subsequently modified within the group of theories that are distinguished below as “combinatory” categorial grammars. These two systems differ slightly in the way they denote directionality, as illustrated in the following categories for the transitive verb *sees*:¹

¹Both notations reflect the assumption that multi-argument functions like transitive verbs are “curried”. Other notations allow “flat” multi-argument functions. Under an equivalence noted by Schönfinkel 1924, the assumption is merely one of notational con-

- (1) a. Lambek: $sees := (np \backslash s) / np$
b. Combinatory CG: $sees := (S \backslash NP) / NP$

Lambek’s notation encodes directionality in the slash itself, forward slash, /, indicating a rightward argument and backward slash \ indicating a leftward argument. However, for reasons which will become apparent when we turn to examine the Lambek calculus in detail, Lambek chose to make leftward arguments appear to the left of their (backward) slash, while rightward arguments appeared to the right of their (forward) slash.² This notation has the disadvantage of not having a consistent left to right order of domain and range. It is therefore hard for the human reader to interpret categories in this notation. The reader may judge this difficulty for themselves by noting how long it takes them to decide whether the two functions written $(a/b) \backslash (c/d)$ and $(d \backslash c) / (b \backslash a)$ do or do not have the same semantic type.³ This property makes life difficult, for example, for linguists whose concern is to compare the syntactic behaviour of semantically related verbs across languages with different base constituent orders.

It was for this last reason that Dowty and the present author proposed an alternative notation with a consistent left-to-right order of range and domain of the function. In this notation, arguments always appear to the right of the slash, and results to the left.⁴ A rightward-leaning slash means that the argument in question is to the right, a leftward-leaning slash, that it is to the left. The first argument of a complex function category is always the rightmost category, the second argument the next rightmost, and so on, and

venience. The categories as shown are simplified by the omission of number and person agreement specifications. In common with most theories, we assume that the categories here represented as atomic NPs are in fact feature bundles including agreement features which must unify with corresponding features of their arguments.

²I have used np as the type of NPs in Lambek’s notation, rather than n , as in the original.

³They do: the semantic type is $(b \rightarrow a) \rightarrow (d \rightarrow c)$.

⁴In this respect it harks back to Bar-Hillel’s original 1953 notation. Bar Hillel’s own version was particularly cumbersome, and in 1960a he abandoned it in favour of the Lambek notation. However, Lyons 1968 offers an extremely elegant version, in which directionality is marked by superior arrows, as in \overrightarrow{np} , \overleftarrow{np} , so that the English transitive verb can be written $(s / \overrightarrow{np}) / \overleftarrow{np}$. A related notation is used by Huck 1988. Unfortunately, until all linguists are equipped with advanced computer typesetting facilities, this does not seem to be a practicable alternative.

the leftmost basic category is always the result. It is therefore obvious in this notation that the two categories instanced in the last paragraph, which are now written $(C/D)\backslash(A/B)$ and $(C\backslash D)/(A\backslash B)$, have the same semantic type, since the categories are identical apart from the slashes.

All the notations illustrated in 1 capture the same basic syntactic facts concerning English transitive sentences as the familiar production rules in 2:

- (2) $S \rightarrow NP VP$
 $VP \rightarrow TV NP$
 $TV \rightarrow \text{sees}$

That is to say that in order to permit parallel context-free derivations we need only include the following pair of rules of functional application, allowing functor categories to combine with arguments (the rules are given in both notations):

- (3) FUNCTIONAL APPLICATION: FUNCTIONAL APPLICATION:
 a. $x/y \quad y \Rightarrow x$ a. $X/Y \quad Y \Rightarrow X$
 b. $y \quad y\backslash x \Rightarrow x$ b. $Y \quad X\backslash Y \Rightarrow X$
 a. Lambek b. Combinatory

These rules have the form of very general binary PS rule schemata. Clearly what we have here is a context free grammar which happens to be written in the accepting, rather than the producing, direction, and in which there has been a transfer of the major burden of specifying particular grammars from the PS rules to the lexicon. (CG and CFPSG were shown by Bar-Hillel et al. 1960b to be weakly equivalent). While it is now convenient to write derivations in both notations as follows, they are clearly just familiar phrase-structure "trees" (except that they have the leaves at the top, as is fitting).

- (4) Gilbert sees George Gilbert sees George

 np (np\s)/np np NP (S\NP)/NP NP
 ----->
 np\s S\NP
 -----< -----<
 s S
 a. Lambek b. Combinatory

(The operation of combination by the application rules is indicated by an underline annotated with a rightward or leftward arrow.) It will be clear at this point that, Lambek's notation has the very attractive property of allowing all "cancellations" under the rules of functional application to be with adjacent symbols. This elegant property is preserved under the generalisation to other combinatory operations permitted by the generalisation to the Lambek calculus. (However, we shall see that it cannot be preserved under the full range of combinatory operations that have been claimed by categorial grammarians to be required for natural languages.)

Grammars of this kind have a number of features that make them attractive as an alternative to the more familiar phrase structure grammars. The first is that they avoid the duplication in syntax of the subcategorisation information that must be explicit in the lexicon anyway. The second is that the lexical syntactic categories are clearly very directly related to their semantics. This last property has always made categorial grammars particularly attractive to formal semanticists, who have naturally been reluctant to give up the belief that natural language syntax must be as directly related to its semantics as that of arithmetic, algebra, or the predicate calculus, despite frequent accusations of extreme over-optimism from linguistic syntacticians.

At the very time Bar-Hillel and Lambek were developing the earliest categorial grammars, Chomsky was developing an argument that many phenomena in natural languages could not be naturally expressed using context free grammars of any kind, if indeed they could be captured at all. It is therefore important to ask how this pure context-free core can be generalised to cope with the full range of constructions found in natural language.

§2 EARLY GENERALISATIONS OF CATEGORIAL GRAMMAR

We should distinguish three types of proposal that came from categorial grammarians in response to this challenge. The first was simply to take over the Chomskian apparatus of transformations, replacing his CFPS base grammar with a pure CF categorial grammar. This proposal was influentially advanced by Lyons 1968, p.227 ff., p.327 ff., and endorsed by Lewis 1972, p.22. Lyons's arguments were based on the advantages of a categorial base for capturing the word-order generalisations associated with the then nascent \bar{X} -theory (which were explored in categorial terms by Flynn 1983), and were

prescient of the subsequent tendency of Chomsky's theory towards lexicalism and a diminished role for PS rules. However, there was increasing awareness at this time that transformational rules themselves needed replacing by some more constrained formal mechanism, and this awareness gave rise to several more radical categorially-based alternative proposals.

The paper in which Lewis endorses Lyons's proposal for a categorially based transformational grammar is in fact only peripherally concerned with syntax. Its more central concern is quantifier scope, which motivates Lewis to introduce a transformational rule which we would nowadays recognise as "Quantifier Raising", complete with the suggestion that this rule should operate "*beneath* ... the most ordinary level of deep structure" – that is at what we would now call the level of logical form (1972, p.198). However, Lewis's account also involves an abstraction operator equivalent to Church's λ , in the form of Ajdukiewicz's operator $\hat{\cdot}$. Implicit in the general approach of Montague 1970, p.223, n.2 (though not in the practice of Montague 1973), and explicit in the approach of Keenan 1971, Venneman (cf. Bartsch and Venneman 1972), and the " λ -categorial" grammars of Cresswell 1973, p.7 and von Stechow 1974, is the proposal that with the abstraction operator there is no need for independent movement transformations at all. Compositional interpretations can be assembled on the basis of surface grammar augmented by the completely general variable-binding operation of λ -abstraction, a proposal that was implicit in Ajdukiewicz.

This bold approach was also prescient of coming moves within the transformational mainstream, anticipating (and possibly, via work in Montague Grammar helping to precipitate) the move in Chomsky's theory to small numbers of general purpose movement transformations, perhaps confined to a single most general rule "*move α* ", and the realisation that all such "movements", even those involving *Wh*-elements and their traces, could be regarded as base-generated. (O'Grady 1991, who combines a categorial base with rules for combining non-adjacent elements, can be seen as continuing this tradition within CG). However, by the same token, the essential equivalence between λ -abstraction ("bind a variable anywhere in the domain") and *move- α* ("co-index any items in the domain") means that the abstraction device is potentially very unconstrained, as Cresswell recognised (1973, p.224-227). The approach remains immensely productive in the semantic domain. It remains to be seen whether there is any explanatory advantage

inherent in the syntactic aspects of λ -categorial grammars. Nevertheless, it has made the important contributions of providing a clear and simple interpretation for the notion of movement itself, which might otherwise have appeared semantically unmotivated, and of having directly led, via the work of Emmon Bach, to the third, most recent, and most radical group of proposals for generalising pure categorial grammar.

As a part of a wider tendency at the time to seek low-power alternatives to transformations, there during the '70s a number of proposals for augmenting categorial grammar with additional operations for combining categories, over and above the original rules of functional application. In contrast to the λ -categorial approach, these operations were less general than the abstraction operator of λ -categorial grammar, the chief restriction being that, like the application rules themselves, these operations were confined to the combination of non-empty string-adjacent entities, and were dependent on the directionality of those entities. These proposals had an important historical precedent in work by Lambek 1958.

Lambek's short paper can be seen as making two quite separate points. The first was that a number of simple functional operations, importantly including functional composition and type-raising, looked as though they were directly reflected in natural syntax. His second point was that these very operations, together with an infinite set of related ones, could be generated as theorems of a quite small set of axioms and inference rules. In this he drew on even earlier traditions of natural deduction in the work of Gentzen (1934, cf. Kleene 1952, Ch.15), and the analogy drawn between logical implication and functional types by Curry (e.g. Curry and Feys 1958), which he deployed in an important proof of decidability for his syntactic calculus. The effect was to define this version of categorial grammar as a restricted logic.

These two proposals can be seen as reflected in two distinct styles of modern categorial grammar. On the one hand, there is a group of linguists who argue that the addition of a few semantically simple primitive combinatory operations like functional composition yields grammars that capture linguistic generalisations. Sometimes these operations are individual theorems of the Lambek calculus, and sometimes they are not. These theorists are typically not concerned with the question of whether their operations can be further reduced to an axiomatic calculus or not (although they are of

course deeply concerned, as any linguist must be, with the degrees of freedom that their rules exhibit, and the automata-theoretic power implicit in their theory). In this respect they are close in spirit to the *semantic* tradition in formal logic.

The other modern school of categorial grammarians are more concerned to identify additional sets of axiom-schemata and inference rules that define other syntactic calculi, primarily as a way of looking at relations among logics, particularly intuitionistic or constructive ones, including modal logics, linear logic, and type-theory. The relation of such logics to natural grammars is often not the central issue. These authors are closer to the *proof-theoretic* tradition in formal logic.

It will be easiest to discuss Lambek’s original proposal in the light of these more recent developments. In adopting this narrative tactic, we recapitulate the history of the subject, for the significance of Lambek’s proposals was not appreciated at the time, and his paper was largely forgotten until rediscovery of many of its principles in the ’70s and early ’80s by Geach, Bach, Buszkowski, and others.

§3 MODERN CATEGORIAL THEORIES OF GRAMMAR

This section begins by examining the “Combinatory” style of categorial grammar, before returning to the “Lambek” style including Lambek’s original proposal. Each of these subsections ends with a brief discussion of the automata-theoretic power inherent in each system. It is convenient to further distinguish certain theories within both frameworks that are mainly concerned with the semantics of quantifier scope, rather than with purely syntactic phenomena. This work is discussed in a third subsection.

¶3.1 “COMBINATORY” CATEGORIAL GRAMMARS

A major impulse behind the development of generalised categorial grammars in this period was an attempt to account for the apparent vagaries of coordinate constructions, and to bring them under the same principles as other unbounded phenomena, such as relativisation.

To begin to extend categorial grammar to cope with coordination we need

a rule, or rather a family of rules, of something like the following form:⁵

- (5) COORDINATION RULE (<&>):
 $X' \text{ conj } X'' \Rightarrow X'''$

This rule captures the ancient intuition that *coordination is an operation which maps two constituents of like type onto a constituent of the same type*. That is, X', X'' and X''' are categories of the same type X but different interpretations, and the rule is a schema over a finite set of rules whose semantics we shall ignore here.⁶ Given such a rule or rule schema, derivations like the following are permitted:

- (6) Harry cooked and ate apples
- NP (S\NP)/NP conj (S\NP)/NP NP
- <&>
- (S\NP)/NP
- >
- S\NP
- <
- S

The driving force behind much of the early development of the theory was the assumption that *all* coordination should be this simple – that is, combinations of *constituents* without the intervention of deletion, movement, or equivalent unbounded coindexing rules (cf. Partee & Rooth 1983, Keenan & Faltz 1985, Zwarts 1986, among others.) Sentences like the following are among the very simplest to challenge this assumption, since they involve the coordination of substrings that are *not* normally regarded as constituents:

- (7) a. Harry cooked, and *might eat*, some apples
b. Harry cooked, and *Mary ate*, some apples
c. Harry will copy, and *file without reading*, some articles concerning Swahili.

⁵The rule as given is a simplification, in that it does not represent the “prepositional” or “proclitic” character of the English conjunctions, which associate to the right as the above category does.

⁶There is a temptation to handle coordination by assigning categories like the following to conjunctions like *and*:

i. $and := (S \setminus S)/S$

We shall see later why this will not work, for reasons first noted by Lambek 1961, p.167.

The problem can be solved by adding a small number of operations that combine functions in advance of their arguments. Curry and Feys 1958 offer a mathematics for capturing applicative systems equivalent to the λ -calculi entirely in terms of such operators, for which they coined the term *Combinator* – hence the term “Combinatory” categorial grammars.⁷

AN ASIDE ON COMBINATORS: A combinator is an operation upon sequences of functions and/or arguments. Thus, any (prefixed) term of the λ -calculus is a combinator. We shall be interesting in combinators that correspond to some particularly simple λ -terms. For example:

$$\begin{aligned}
 (8) \quad & a. \mathbf{I} \quad \equiv \lambda x[x] \\
 & b. \mathbf{K}y \quad \equiv \lambda x[y] \\
 & c. \mathbf{T}x \quad \equiv \lambda F[Fx] \\
 & d. \mathbf{B}FG \quad \equiv \lambda x[F(Gx)] \\
 & e. \mathbf{C}Fy \quad \equiv \lambda x[Fxy] \\
 & f. \mathbf{W}F \quad \equiv \lambda x[Fxx] \\
 & g. \mathbf{S}FG \quad \equiv \lambda x[Fx(Gx)] \\
 & h. \mathbf{\Phi}HFG \equiv \lambda x[H(Fx)(Gx)] \\
 & \text{where } x \text{ is not free in } F, G, H, y.
 \end{aligned}$$

(A convention of “left-associativity” is assumed here, according to which expressions like $\mathbf{B}FG$ are implicitly bracketed as $(\mathbf{B}F)G$. Concatenation as in $\mathbf{T}x$ denotes functional application of \mathbf{T} to x .)

The above are equivalences, not definitions of the combinators. The combinators themselves can be taken as primitives, and used to define a range of *applicative systems*, that is systems which express the two notions of *application* of functions to arguments, and *abstraction* or definitions of functions in terms of other functions. In particular, suprisingly small collections of combinators can be used to as primitives to define systems equivalent to various forms of the λ -calculus, entirely without the the use of bound variables and the binding operator λ .⁸

⁷Curry himself discussed the relation of applicative systems to grammars in 1961, proposing, albeit in programmatic terms, a monostratal alternative to transformational grammar (pp. 65-66). One categorial theory acknowledging direct descent from this paper is that of Dahl 1977.

⁸Curry and Feys 1958, Ch.5, (written by W. Craig) – remains the most accessible introduction to Combinatory Logic. A very attractive alternative is provided by Smullyan’s

It is usual to show constructively that a given system of combinators is equivalent in expressive power to one of the λ -calculi, by providing an algorithm that will map any expression of the latter into an equivalent combinatory expression. One of the smallest and most elegant sets that is complete in this way consists of three combinators, **I**, **K** and the familiar **S** combinator. The algorithm can be represented as three cases, as follows

$$(9) \quad \begin{aligned} \lambda x[x] &= \mathbf{I} \\ \lambda x[y] &= \mathbf{K}y \\ \lambda x[AB] &= \mathbf{S}\lambda x[A]\lambda x[B] \end{aligned}$$

The first two steps represent the two ground conditions of abstracting over the variable itself and abstracting over any other atom. The third step says that abstracting over a compound term consisting of the application of a function term A to an argument term B is equivalent to applying the combinator **S** to the result of abstracting over the function and over the argument. (Given the earlier definition of **S**, it is easy to verify that this equivalence holds.) Since the combinator **I** can in turn be defined in terms of the other two combinators (as **SKK**), the algorithm is often referred to as the **SK** algorithm. It is attributed by Curry and Feys to Rosser. It is obvious that the algorithm is complete, in the sense that it will deliver a combinatory equivalent of any λ term.⁹ Other algorithms can be devised using others of the combinators identified earlier (fortunately, some yield less cumbersome combinatory expressions than the **SK** algorithm).

BTS COMBINATORY CATEGORIAL GRAMMAR: One combinatory generalisation of categorial grammar adds exactly three further classes of combinatory rule to the context-free core. Since two of these types of rule – namely composition and type-raising – have been at least implicit in the majority of combinatory generalisations of categorial grammars, and since a third operation is provably necessary, we will take this system as the canonical exemplar,

1985 *To Mock a Mockingbird*. Smullyan's system of combinators, which are named after birds, resembles the combinatory system described in the next section in the central role it accords to Bluebirds and Thrushes. However, the reader should be aware that Curry and Smullyan are concerned with *untyped* combinators. While many of their results transfer to the typed combinators implicit in the linguistic proposal, "paradoxical" combinators like Curry's **Y** and Smullyan's Mockingbird are not valid in typed systems.

⁹Curry and Feys give the formal proof.

comparing it later to a number of variants and alternatives.¹⁰ The combinatory rules have the effect of making such substrings into grammatical constituents in the fullest sense of the term, complete with an appropriate and fully compositional semantics. All of them adhere to the following restrictive assumption:

- (10) THE PRINCIPLE OF ADJACENCY: Combinatory rules may only apply to entities which are linguistically realised and adjacent.

The first such rule-type is motivated by examples like 7a, above. Rules of functional composition allow functional categories like *might* to combine with functions into their argument categories, such as *eat* to produce non-standard constituents corresponding to such strings as *might eat*. The rule required here (and the most commonly used functional composition rule in English) is written as follows:

- (11) FORWARD COMPOSITION (\Rightarrow_B):
 $X/Y \ Y/Z \Rightarrow_B \ X/Z$

The rule permits the following derivation for example 7a:

- (12)
- | | | | | | | |
|-------|-----------|------|-----------|-----------|-----------|-----|
| Harry | cooked | and | might | eat | the beans | |
| | | | | | | |
| NP | (S\NP)/NP | conj | (S\NP)/VP | VP/NP | NP | |
| | | | | | | >B |
| | | | | (S\NP)/NP | | |
| | | | | | | <&> |
| | | | | (S\NP)/NP | | |
| | | | | | | > |
| | | | | S\NP | | |
| | | | | | | < |
| S | | | | | | |

It is important to observe that, because of the isomorphism that CG embodies between categories and semantic types, this rule is also *semantic* functional composition. That is, if the interpretations of the two categories on

¹⁰This variety, with whose development the present author has been associated is sometimes referred to as CCG (for Combinatory Categorical Grammar), although it is only one of the possible combinatory versions of CG.

the left of the arrow in 11 are respectively F and G , then the interpretation of the category on the right must be the composition of F and G . Composition corresponds to Curry's composition combinator, which he called **B**, defined earlier as 8d.¹¹ Hence, the combinatory rule and its application in the derivation are indexed as $\triangleright B$ because it is a rule in which the main functor is rightward-looking, and has composition as its semantics. Hence also, the formalism guarantees without further stipulation that this operation will compose the interpretations, as well as the syntactic functional types. We will defer formal discussion of this point, but it should be obvious that if we know the mapping from VP interpretations to predicate interpretations that constitutes the interpretation of *might*, and we know the mapping from NP interpretations to VP interpretations corresponding to the interpretation of *eat*, then we know everything necessary to define their composition, the interpretation of the non-standard constituent *might eat*.

The result of the composition has the same syntactic and semantic type as a transitive verb, so when it is applied to an object and a subject, it is guaranteed to yield exactly the same interpretation for the sentence *Harry might eat some beans* as we would have obtained without the introduction of this rule. This non-standard verb *might eat* is now a constituent in every sense of the word. It can therefore coordinate with other transitive verbs like *cooked* and take part in derivations like 12. Since this derivation is in every other respect just like the derivation in 6, it too is guaranteed to give a semantically correct result.

Examples like the following, in which a similar substring is coordinated with a *di*-transitive verb, require a generalisation of composition proposed by Ades and the present author 1982:¹²

- (13) I *will offer*, and [*may*]_{(S\NP)/VP} [*sell*]_{(VP/PP)/NP},
 my 1959 pink cadillac to my favourite brother-in-law

To compose the modals with the multiple-argument verbs, we need the following relative of rule 11:

¹¹Curry 1958, p.184, fn., notes that he called the operation **B** because it occurs prominently in the word "substitution", and because the names **S** and **C** were already spoken for. The operation is Smullyan's Bluebird.

¹²These sentences are better when one of the extractions is a relativisation (see below), as in *the man to whom I will offer, and may sell, my 1959 pink cadillac*.

- (14) FORWARD COMPOSITION ($\triangleright B2$):
 $X/Y \ (Y/Z)/W \Rightarrow_{B^2} (X/Z)/W$

This corresponds in combinatory terms to an instance B^2 of the generalisation from B to B^n (cf. Curry & Feys 1958, p.165 and 185). We can assume, at least for English, that n is bounded by the highest valency in the lexicon, which is about 4.

The second novel kind of rule that is imported under the combinatory generalisation is motivated by examples like 7b above, repeated here:

- (15) Harry cooked, and Mary ate, some apples

If the assumption is to be maintained that everything that can coordinate is a constituent formed without deletion or movement, then *Harry* and *cooked* must also be able to combine to yield a constituent of type S/NP , which can combine with objects to its right. The way this is brought about is by adding rules of type-raising like the following to the system:

- (16) FORWARD TYPE-RAISING ($\triangleright T$):
 $Y \Rightarrow_T X/(X \backslash Y)$

This rule makes the subject NP into a function over predicates. Subjects can therefore compose with functions *into* predicates – that is, with transitive verbs, as in the following derivation for 15:¹³

- (17)
- | | | | | | |
|----------|-----------|-------|----------|-----------|-------------|
| Harry | cooked | and | Mary | ate | some apples |
| ----- | ----- | ----- | ----- | ----- | ----- |
| NP | (S\NP)/NP | conj | NP | (S\NP)/NP | NP |
| ----->T | | | ----->T | | |
| S/(S\NP) | | | S/(S\NP) | | |
| ----->B | | | ----->B | | |
| S/NP | | | S/NP | | |
| | -----<&> | | | | |
| | S/NP | | | | |
| | -----> | | | | |
| | | | S | | |

¹³Agreement is ignored as usual.

Type-raising corresponds semantically to the combinator **T**, defined at 8c.¹⁴ We shall see later that type-raising is quite general in its application to NPs, and that it should be regarded as an operation of the lexicon, rather than syntax, under which all types corresponding to functions into *NP* (etc.) are replaced by functions into the raised categor(ies). However, for expository simplicity we shall continue to show it in derivations, indexing the rule as **>T**. When the raised category composes with the transitive verb, the result is guaranteed to be a function which, when it reduces with an object *some apples*, will yield the same interpretation that we would have obtained from the traditional derivation. This interpretation might be written as follows:

(18) *cook' apples' harry'*

(Here again we use a convention of “left associativity”, so that the above applicative expression is equivalent to *(cook' apples') harry'*.) It is important to notice that it is at the level of the interpretation that traditional constituents like the VP, and relations such as *c*-command, continue to be embodied. This is an important observation, to which we return below, since as far as surface structure goes, both have now been compromised.

Of course, the same facts guarantee that the coordinate example above will deliver an appropriate interpretation.

The third and final variety of combinatory rule is motivated by examples like 7c, repeated here:

(19) Harry will copy, and file without reading, some articles concerning Swahili

Under the simple assumption with which we began, that only like *constituents* can conjoin, the substring *file without reading* must be a constituent formed without movement or deletion. What is more, it must be a constituent of the same type as a transitive verb, *VP/NP*, since that is what it coordinates with. It follows that the grammar of English must include the following operation, first proposed by Szabolsci (1983, 1987b):¹⁵

¹⁴The rule was called **C*** by Curry, and is Smullyan’s Thrush. Type-raising is of course widely used in Montagovian semantics.

¹⁵The name “substitution” was proposed for the combinator **S** in homage to Curry’s explanation (referred to in an earlier footnote) of his choice of the name **B** as deriving

- (20) BACKWARD CROSSED SUBSTITUTION ($\langle Sx \rangle$)
 $Y/Z \ (X \backslash Y)/Z \Rightarrow_S X/Z$

This rule permits the following derivation for the sentence:¹⁶

- (21) Harry will copy and file without reading, some articles
- | | | | | | | |
|-------|-------|--------|----------|---------------|----------|-------|
| ----- | ----- | ----- | ----- | ----- | ----- | ----- |
| S/VP | VP/NP | conj | VP/NP | (VP\VP)/VPing | VPing/NP | NP |
| | | | | ----->B | | |
| | | | | (VP\VP)/NP | | |
| | | | | -----<Sx | | |
| | | | VP/NP | | | |
| | | | -----<&> | | | |
| | | | VP/NP | | | |
| | | | ----->B | | | |
| | | S/NP | | | | |
| | | -----> | | | | |
| | | S | | | | |

It is important to notice that the crucial rule resembles a generalised form of functional composition, but that it *mixes* the directionality of the functors, combining a leftward functor over VP with a rightward function into VP . We must therefore predict that other combinatory rules, such as composition, must also have such “crossed” instances. Such rules are not valid in the Lambek calculus.

Like the other combinatory rules, the substitution rule combines the interpretations of categories as well as their syntactic categories. Its semantics is given by the combinator S , defined at 8g. It follows that if the constituent *file without reading* is combined with an object *some articles* on the right, and then combined with *Harry will*_{S/VP}, it will yield a correct interpretation. It also follows that a similarly correct interpretation will be produced for the coordinate sentence 19.

from this word, and because S is the general form of the operation of which B is a special case. Schönfinkel 1924 called it *Verschmelzung*, or “fusion”. Kaplan 1975 called it “composition” (!), and Szabolcsi 1983 called it “connection”. It is Smullyan’s Starling. The family of combinators associated with the coordination rule 5, whose semantics has so far been ignored, was called Φ by Curry, and is closely related to S

¹⁶Infinitival and gerundival predicate categories are abbreviated as VP and VP_{ing} , and NPs are shown as ground types.

These three classes of rule – composition, type-raising, and substitution – constitute the entire inventory of combinatory rule-types that this version of combinatory CG adds to pure categorial grammar. They are limited by two general principles, in addition to the Principle of Adjacency 10. They are the following:

- (22) THE PRINCIPLE OF DIRECTIONAL CONSISTENCY: All syntactic combinatory rules must be consistent with the directionality of the principal function.
- (23) THE PRINCIPLE OF DIRECTIONAL INHERITANCE: If the category that results from the application of a combinatory rule is a function category, then the slash defining directionality for a given argument in that category will be the same as the one defining directionality for the corresponding argument(s) in the input function(s)

Together they amount to a simple statement that *combinatory rules may not contradict the directionality specified in the lexicon*. They drastically limit the possible composition and substitution rules to exactly four instances each. It seems likely that these principles follow from the fact that directionality is as much a property of *arguments* as is their syntactic type. This position is closely related to Kayne’s 1984 notion of *directionality of government*.

The inclusion of this particular set of operations makes a large number of correct predictions. For example, once we have seen fit to introduce the forward rule of composition and the forward rule of type raising into the grammar of English, we do not increase the degrees of freedom in the theory any further by introducing the corresponding *backward* rules. Thus the existence of the following coordinate construction is predicted without further stipulation, as noted by Dowty 1988:¹⁷

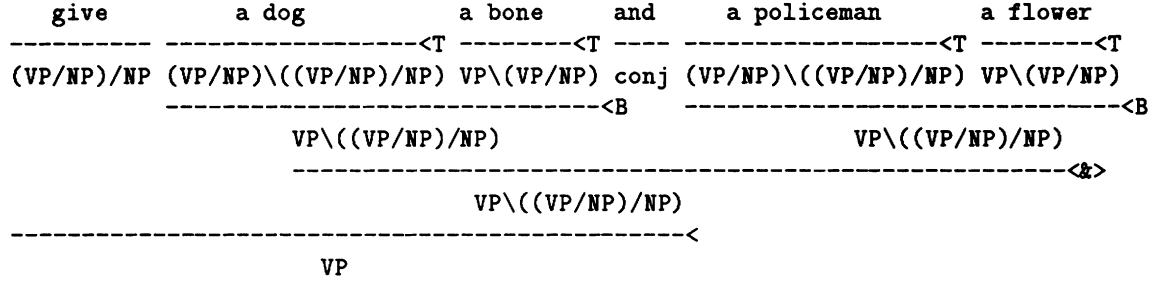
¹⁷The two rules that are involved are the following (Lambek-provable) rules:

- (i) BACKWARD TYPE-RAISING (<T):

$$Y \Rightarrow_{\mathbf{T}} X \backslash (X/Y)$$
- (ii) BACKWARD COMPOSITION (<B):

$$Y \backslash Z \quad X \backslash Y \Rightarrow_{\mathbf{B}} X \backslash Z$$

(24)



This and other related examples, which notoriously present considerable problems for other grammatical frameworks (cf. Hudson 1982), are extensively discussed by Dowty and others, and constitute strong evidence in support of the decision to take type raising and composition as primitives of grammar.¹⁸

The analysis also immediately entails that the dependencies engendered by coordination will be unbounded, and free in general to apply across clause boundaries. For example, all of the following examples parallel to the triple 7 with which we began the section are immediately accepted, without any further addition to the grammar whatsoever:

- (25) a. Harry cooked, and *expects that Mary will eat*, some apples
 b. Harry cooked, and *Fred expects that Mary will eat*, some apples
 c. Harry cooked, and *Fred expects that Mary will eat without enjoying*,
 some apples that they found lying around in the kitchen.

Moreover, if we assume that nominative and accusative relative pronouns have the following categories, (which simply follow from the fact that they are functions from properties to noun modifiers), then we also accept the relative clauses in 27, below:

- (26) a. *who/that/which* := $(N \setminus N)/(S \setminus NP)$
 b. *who(m)/that/which* := $(N \setminus N)/(S/NP)$

The inclusion of (i) suggests that type raising is a general process that should apply to all categories whose range is NP in the lexicon. We pass over the question of how this can be done without enlarging the lexicon unduly.

¹⁸However, Oehrle 1987, 1988a and Wood 1988 offer important alternative analyses for examples like 24 in terms of operations related to Lambek's 1958 product operator.

word order, including the possibility of “scrambling” arguments across clause boundaries. It is therefore assumed in this version of combinatory categorial grammar that languages are free to restrict such rules to certain categories, or even to exclude them entirely.

One of the most interesting observations to arise from the movement analysis of relatives is the observation that there are a number of striking limitations on relativisation. The exceptions fall into two broad classes. The first is a class of constraints relating to asymmetries with respect to extraction between subjects and objects. This class of exceptions have been related to the “empty category principle” (ECP) of GB. In the terms of the combinatory theory, this constraint arises as a special case of a more general corollary of the theory, namely that arguments of different directionality require different combinatory rules to apply if they are to extract, as inspection of the following examples will reveal. The possibility for such asymmetries to exist in SVO languages because of the exclusion of the latter non-direction preserving rule is therefore open.

- (30) a. (a man whom) [I think that]_{S/S} [Mary likes]_{S/NP}
 b. *(a man whom) [I think that]_{S/S} [likes Mary]_{S\NP}

Indeed, a language like English *must* limit or exclude this rule if it is to remain configurational.²¹

The second class is that of so-called “island constraints”, which have been related to the principle of “subjacency”. The fact that adjuncts and NPs are in general islands follows from the assumption that the former are backward modifiers, and that type raising is lexical and restricted to categories which are arguments of verbs, such as NPs. This can be seen from the categories in the following unacceptable examples:

- (31) a. * a book [which]_{(N\N)/(S/NP)} [I will]_{S/VP} [walk]_{VP} [without reading]_{(VP\VP)/NP}
 b. * a book [which]_{(N\N)/(S/NP)} [I met]_{S/NP} [a man who wrote]_{(S\NP)/NP}

The possibility of exceptions to the island status of NPs and adjuncts, and their equally notorious dependence on lexical content and such semantically

²¹The question of whether the grammar of *non*-configurational languages can be correctly ascribed to the free play of such rules is an open research question.

related properties as definiteness and quantification, can be explained on the assumption that verbs can be selectively type-raised over such adjuncts, and lexicalised. Thus the possibility of exceptions like the following (and the generally uncertain judgements that are associated with sentences involving subadjacency violations) are also explained:²²

- (32) a. ?a man who I painted a picture of
 b. ?an article which I wrote my thesis without being aware of.

The subadjacency constraints are treated at length by Szabolcsi and Zwarts 1990 and Hepple 1990.

Other theories on this branch of the categorial family have proposed the inclusion of further combinators, and/or the exclusion of one or more of the above. Perhaps the first of the modern combinatory theories, that of Bach 1979, 1980, proposed an account of certain bounded constructions, including passive and control, by a “wrapping” operation which combined functions with their second argument in advance of their first, an analysis which is extended by Dowty 1982, Jacobson 1987, 1990, Huck 1988, Steele 1990, and Hoeksema 1991. Such operations are related to (but not identical to) the “associativity” family of theorems of the Lambek calculus (Lambek 1958, and below). They are also closely related to the **C** or “commuting” family of combinators. They can also be simulated by, or defined in terms of, the composition and type lifting combinators, as we saw for in the last example. Shaumyan, 1977, Desclés et al. 1986, and Szabolcsi 1987a also implicate Curry’s combinator **W** in their analyses of reflexives. Cormack’s 1986 and Jacobson’s 1990 theory of related constructions exploits functional composition in accounting for raising, equi and the like, with important implications for the treatment of VP anaphora.

Since all of the above constructions are bounded, the theories in question can be viewed as combinatory theories of the lexicon and of lexical morphology (cf. Hoeksema 1985, although see Bach 1979, 1980 and Jacobson 1990 for arguments against too simplistic an interpretation of this view). To that extent, the above theories are close relatives of the theories of Keenan

²²The suggestion that subadjacency and its exceptions are lexical and therefore ultimately semantic in origin is closely related to the unification of notions of subadjacency and government via the notion of “barrier” in Chomsky 1986, p.10-16.

& Faltz 1985, and to the theory of Shaumyan 1977. All of these theories embody related sets of operations in lexical semantics. Shaumyan in particular explicitly identifies these operations with a very full range of Curry’s combinators.

POWER OF COMBINATORY GRAMMARS: One may ask at this point what the power of such grammars is. We have already seen that collections of combinators as small as the pair **SK** may have the full expressive power of the lambda calculus. **BCWI** and **BCSI** are also implicitly shown by Curry and Feys to be equivalent to the λ_I -calculus – that is, the lambda calculus without vacuous abstraction. The present system of (typed) **BST** is also essentially equivalent to the (simply typed) λ_I -calculus, although technically we may need to include the ground case of **I** where its argument is a single variable as a special case.²³ This equivalence means that any restrictiveness that inheres to the theory in automata-theoretic terms stems from the directional sensitivity inherent in the lexicon and in the Principles of Consistency 22 and Inheritance 23 alone.

Joshi, Vijay-Shankar and Weir 1987 have recently shown that a number of “mildly non-context-free” grammar formalisms including Joshi’s Tree-Adjunction Grammars (TAG), Pollard’s Head Grammars (HG), and the version of combinatory categorial grammar sketched here can be mapped onto Linear Indexed Grammars.²⁴ Indexed grammars are grammars which, when represented as phrase structure rewriting systems, allow symbols on both sides of a production to be associated with features whose values are *stacks*, or unbounded push-down stores. We can represent such rules as follows, where the notation $[...]$ represents a stack-valued feature under a convention that the top of the stack is to the left:

$$(33) \quad \alpha_{[...]} \rightarrow W_1 \beta_{[...]} W_2$$

²³No expression of the general type $a \rightarrow a$ can be formulated in terms of **BST**. However, an expression of type $(a \rightarrow b) \rightarrow (a \rightarrow b)$ is typeable.

²⁴There is considerable recent convergence between these theories. “Lexicalised” TAGs (LTAG, Schabes, Abeillé & Joshi 1988) and “Head-driven” Phrase-structure Grammars (HPSG, Pollard & Sag 1987) are regarded by their proponents as closely related to Categorical Grammar. The version of categorial grammar known as CCG has in turn been considerably influenced recently by work in TAG and HPSG.

Such rules have the effect of passing a feature from a parent α to one or more daughters β which may encode long range dependencies. The rules are allowed to make two kinds of modifications to the stack value: an extra item may be “pushed” onto the top of the stack, or the topmost item that is already on the stack may be removed. These two types of rule can be represented as similar schemata, as follows:

$$(34) \quad \begin{array}{lcl} \text{“pushing”} & \alpha_{[\dots]} & \rightarrow W_1 \beta_{[i,\dots]} W_2 \\ \text{“popping”} & \alpha_{[i,\dots]} & \rightarrow W_1 \beta_{[\dots]} W_2 \end{array}$$

In general, indexed grammars may include rules which pass stack-valued features to more than one daughter. The most restrictive class of indexed grammars, Linear Indexed Grammars, allows the stack valued feature to pass to only one daughter.

It is easy to show that linear indexed grammars can very directly capture such non-context-free grammars as $a^n b^n c^n$

There is an obvious mapping between functions of n arguments a_1 to a_n into a category α and indexed grammar categories $\alpha_{[a_n, \dots, a_1]}$ bearing an n -deep stack-valued feature. It follows that Combinatory rules can be equally directly represented as indexed productions. For example, the following equivalence holds for the forward composition rule:

$$(35) \quad X/Y \quad Y/Z \Rightarrow X/Z \equiv X'_{[Z,\dots]} \rightarrow X'_{[Y,\dots]} Y_{[Z]}$$

The variable X in the combinatory rule can match any category. It therefore corresponds to an indexed category $X'_{[\dots]}$. Crucially, the stack, represented as $[\dots]$, is only passed to one daughter. The same is true for the substitution rule.

$$(36) \quad Y/Z \quad (X \setminus Y)/Z \Rightarrow X/Z \equiv X_{[Z,\dots]} \rightarrow Y_{[Z]} X_{[Z,Y,\dots]}$$

It is also trivially true for type-raising, although we have seen that this should really be regarded as a lexical rule. It also applies to the rules corresponding to $\mathbf{B}^2, \mathbf{B}^3$ etc, because we claimed that there was a finite limit on the arity of the verbs concerned. However, Joshi et al. point out that a non-finite rule schema corresponding fully generally to \mathbf{B}^n , where n is unbounded, so that

Y corresponds to $Y'_{[...]}$ would *not* be a linear rule, because it would require more than one stack valued feature. Such grammars are of greater expressive power than linear indexed grammars.

The consequences of equivalence to linear indexed grammars are significant, as Joshi et al. show. In particular, linear indexed grammars, by passing the stack to only one branch, allow divide-and-conquer parsing algorithms. As a result, these authors have been able to demonstrate polynomial worst-case limits on the complexity of parsing the version of combinatory CG described above.

¶3.2 LAMBEK-STYLE CATEGORIAL GRAMMARS

Lambek’s original proposal began by offering intuitive motivations for including operations of composition, type-raising, and certain kinds of rebracketing in grammars. All of the operations concerned are, in terms of an earlier definition, *order preserving*. The first two operations are familiar but the last needs some explanation. Lambek notes that a possible “grouping” of the sentence *(John likes)(Jane)* is as shown by the brackets. (He might have used a coordinate sentence as proof, although he did not in fact do so.) He then notes that the following operation would transform a standard transitive verb into a category that could combine with the subject first to yield the desired constituency (the rule is given in Lambek’s own notation, as defined earlier):

$$(37) \quad (np \backslash s) / np \rightarrow np \backslash (s / np)$$

There are two things to note about this operation. One is that it is redundant: that is, its effect of permitting a subject to combine before an object can be achieved by a combination of type-raising and composition, as in example 17. The second is that, while this particular operation is order preserving and stringset-preserving, many superficially similar operations are not. For example, the following rule would not have this property:

$$(38) \quad * (s / np) / np \rightarrow s / (np / np)$$

That is, rebracketing of this kind can only apply across opposite slashes, not across same slashes.

However, Lambek was not proposing to introduce these operations as independent rules. He went on to show in his paper that an infinite set of such-order preserving operations emerged as theorems from a logic defined in terms of a small number of axiom schemata and inference rules. These rules included an identity axiom, associativity axiom schemata, and inference rules of application, abstraction, and transitivity (see Lambek 1958, p.166). The theorems included functional application, the infinite set of order-preserving instances of operations corresponding to the combinators $\mathbf{B}, \mathbf{B}^2, \dots \mathbf{B}^n$, and the order-preserving instances of type raising, \mathbf{T} . They also included the rule shown in 37 and a number of operations of mathematical interest, including the Schönfinkel 1924 equivalence between “flat” and “curried” function-types, and a family of “division rules” including the following:

$$(39) \quad z/y \rightarrow (z/x)/(y/z)$$

The latter is of interest because it was the most important rule in Geach’s proposal (1972, p.485 and see below), for which reason it is often referred to as the “Geach Rule”.²⁵

This last result is also of interest because an elegant alternative axiomatisation of the Lambek calculus in terms of the Geach rule was provided by Zielonka 1981, who dropped Lambek’s associativity axioms, substituting two Geach Rules and two Lifting rules, and dropping the abstraction and transitivity inference rules in favour of two derived inference rules inducing recursion on the domain and range categories of functors. Zielonka’s paper also proved the important result that no finite axiomatisation of the Lambek calculus is possible without the inclusion of some such recursive reduction law. Zielonka’s calculus differs from the original in that the product rule is no longer valid, for which reason it is sometimes identified as the “product-free” Lambek calculus.

The Lambek calculus has the following properties. If a string is accepted on some given lexical assignment, the calculus will allow further derivations corresponding to all possible bracketings of the string. That is, the calculus is “Structurally Complete”. Curiously, while Buszkowski 1982 showed that a version of the calculus restricted to one of the two slash-directions was weakly

²⁵Strictly, it is merely entailed by Geach’s rule as stated, together with a rule of abstraction.

equivalent to context-free grammar, the non-finite-axiomatisation property of the calculus has meant that no proof of the same weak equivalence for the full bi-directional calculus has yet been found. Nevertheless, everyone since Bar-Hillel et al 1960b and Chomsky 1963 has been convinced that the equivalence holds, and Buszkowski 1988a presents a number of partial results which strengthen this conviction.

If we compare the Lambek calculus with the combinatory alternative discussed earlier, then we see the following similarities. Both composition and type-raising are permitted rules in both systems, and both are generalised in ways which can be seen as involving recursive schemata and polymorphism.²⁶ However, there are important divergences between these two branches of the categorial family. The most important is that many of the particular combinatory rules that have been proposed by linguists, while they are *semantically* identical to theorems of the Lambek calculus, are not actually theorems thereof. For example, Bach’s 1979, 1980 rule of “right-wrap”, which shares with Lambek’s rebracketing rule 37 a semantics corresponding to the commuting combinator **C**, is not Lambek-provable. Similarly, examples like 29 have been used to argue for “non-order-preserving” composition rules, which correspond to instances of the combinator **B** that are also unlicensed by the Lambek calculus. It is hard to do without such rules, because their absence prevents all non-peripheral extraction and all non-context-free constructions (see below). Finally, none of the rules that combine arguments of more than one functor, including Geach’s semantic coordination rule, the coordination schema 5, and Szabolsci’s substitution rule 20 are Lambek theorems.

The response of categorial grammarians has been of two kinds. Many linguists have simply continued to take non-Lambek combinatory rules as primitive, the approach discussed in the previous sections. Such authors have placed more importance on the semantic interpretability of the combinatory rules than on further reducibility to axiom systems. In this respect they may be seen as representing a turning away from the proof-theoretic orientation of the Lambek calculus to the alternative, semantic, logical tradition. Others have maintained the proof-theoretic tradition and attempted to identify alternative calculi that have more attractive linguistic properties.

²⁶But note that we have assumed a bound on **B**ⁿ in the combinatory case – cf. section XX below.

Lambek himself was the first to express scepticism concerning the linguistic potential of his calculus, a position that he has maintained to the present day. He noted in 1961, p.167 that, because of the use of a category $(s \backslash s)/s$ for conjunctions, the calculus not only permitted strings like a, below, but also ones like b:

- (40) a. Who walks and talks?
 b. *Who walks and he talks?

The overgeneralisation arises because the conjunction category, having applied to the sentence *He talks* to yield $s \backslash s$, can *compose* with *walks* to yield the predicate category $np \backslash s$. It is exactly this possibility that forces the use of a syncategorematic coordination schema such as 5 in the combinatory approach. However, we have seen that such rules are not Lambek calculus theorems. Lambek's initial reaction was to restrict his original calculus by omitting the associativity axiom, yielding the "non-associative" Lambek calculus. This version, which has not been much used, is unique among extensions of categorial grammar in disallowing composition, which is no longer a theorem.

Other work on the proof-theoretic wing, notably by van Benthem 1986, 1991, Moortgat (whose 1988b book is the most accessible introduction to the area), and Morrill 1988 has attempted to generalise, rather than to restrict, the original calculus. Much of this work has been directed at the possibility of restoring to the calculus one or more of Gentzen's "structural rules", which Lambek's original calculus entirely eschews, and whose omission render it less powerful than full intuitionistic logic. In CG terms, these three rules correspond to *permutation of adjacent categories*, or "interchange", *reduction of two instances of a category to one*, or "contraction", and *vacuous abstraction*, a.k.a "thinning" or "weakening". In combinatory terms, they correspond to the combinator-families **C**, **W**, and **K**. As Lambek points out (1990a, 1990b) a system which allows only the first of these rules corresponds to the linear logic of Girard, while a system which allows only the first two corresponds to the relevance logic **R**_⊥ and the "weak positive implicational calculus" of Church, otherwise known as the λ_I -calculus.

POWER OF LAMBEK-STYLE GRAMMARS: Van Benthem 1986, 1988, 1991 examined the consequences of adding the interchange rule, and showed that

such a calculus is not only structurally complete but “permutation-complete”. That is, if a string is recognised, so are all possible permutations of the string. He shows (1991, p.97) that the this calculus is (in contrast to the original calculus) of greater than context-free power. For example, a lexicon can readily be chosen which accepts the language whose strings contain equal numbers of a’s, b’s, and c’s, which is non-context free. However, Moortgat 1988b, p.118 shows that the theorems of this calculus do not obey the principles of directional consistency 22 and directional inheritance 23 – for example, they include all sixteen possible forms of first-order composition, rather than just four. Moortgat also shows (1988b, pp.92-93) that the mere inclusion in a Lambek-style axiomatisation of slash-crossing composition rules like 28 (which of course are permitted by these principles) is enough to ensure collapse into van Benthem’s permuting calculus. There does not seem to be a natural Lambek-style system in between.²⁷

However, Moortgat does offer a way to generalise the Lambek calculus without engendering collapse into permutation-completeness. He proposes the introduction of new equivalents of slash, including “infixing” slashes, together with axioms and inference rules that discriminate between the slash-types (cf. 1988b, p.111,120), giving the system the character of a “partial” logic. While he shows that one such axiomatisation can be made to entail the generalisations inherent in the principles of consistency and inheritance, it seems likely that many equally simple formulations within the same degrees of freedom would produce much less desirable consequences. Moreover, unless the recursive aspects of this axiom-schematisation can be further constrained limits the such theorems as the composition family \mathbf{B}^n in a similar way to the combinatory alternative, it appears to follow that this calculus is still of greater power than linear indexed grammar.

²⁷Van Benthem has also investigated higher generalisations, such as the calculus including contraction, with and without permutation. While the interest of these systems as logics has been noted already, as far as linguistics goes, the latter system inherits the weakness of the Lambek calculus with respect to non-peripheral extraction, while the former inherits the overgeneralisations of the permuting calculus. Moreover, he shows that all calculi arising from the inclusion of contraction accept only regular languages. This result applies to the calculus that includes both interchange and contraction, which it will be recalled is semantically equivalent to the λ_I -calculus. Not surprisingly, rules that increase the expressive power of a system in semantic terms may catastrophically diminish its weak generative capacity.

In the work of Moortgat, the semantic (combinatory) and proof-theoretic (Lambek-style) traditions of CG come close to convergence. Without the restrictions inherent in the principles of Consistency and Inheritance, both frameworks would collapse. The main difference between the theories is that on the combinatory view the restrictions are built into the axioms and are claimed to follow from first principles, whereas on the Lambek view, the restrictions are imposed as filters.

§4 CATEGORIAL GRAMMARS AND LINGUISTIC SEMANTICS

There are two commonly used notations that make explicit the close relation between syntax and semantics that both combinatory and Lambek-style categorial grammars embody. The first associates with each category a term of the lambda calculus naming its interpretation. The second associates an interpretation with each basic category in a functor, a representation which has the advantage of being directly interpretable via standard term-unification procedures of the kind used in logic programming languages such as Prolog. The same verb *sees* might appear as follows in these notations, which are here shown for the combinatory categories, but which can equally be applied to Lambek categories. In either version it is standard to use a colon to associate syntactic and semantic entities, to use a convention that semantic constants have mnemonic identifiers like *see'* distinguished from variables by primes. For purposes of exposition we will here assume that translations exactly mirror the syntactic category in terms of dominance relations. Thus we adopt a convention of “left associativity” in translations, so that expressions like *see' y x* are equivalent to $(see' y) x$:²⁸

- (41) a. λ -term-based: $sees := (S \backslash NP) / NP : \lambda y \lambda x [see' y x]$
 b. Unification-based: $sees := (S : see' y x \backslash NP : x) / NP : y$

The advantage of the former notation is that the λ -calculus is a highly readable notation for functional entities. Its disadvantage is that we now have to complicate the notation of the combinatory rules to allow the combination of

²⁸It is of course possible to have a variant of notation (a) in which the variables are eliminated, the same category being written as follows:

(i) $(S \backslash NP) / NP : see'$

both parts of the category, as in a, below. This has the effect of weakening the direct relation between syntactic and semantic types, since it suggests we might allow rules in which the syntactic and semantic combinatory operations were *not* identical. In the unification notation b, by contrast, the combinatory rules apply unchanged, and necessarily preserve identity between syntactic and semantic operations, a property which was one of the original attractions of CG.²⁹

- | | |
|--|---|
| <p>(42) FORWARD COMPOSITION:
 $X/Y : f \quad Y/Z : g \Rightarrow X/Z : \lambda x[f(g \ x)]$</p> <p style="text-align: center;">a. λ-term-based</p> | <p>FORWARD COMPOSITION:
 $X/Y \ Y/Z \Rightarrow X/Z$</p> <p style="text-align: center;">b. Unification-based</p> |
|--|---|

Because of their direct expressibility in unification-based programming languages like Prolog, and related special-purpose linguistic programming languages like PATR-II (cf. Shieber 1986), the latter formalism or notational variants thereof are widespread in the computational linguistics literature (cf. Wittenburg 1986; Uszkoreit 1986; Karttunen 1989; Bouma 1987; Zeevat et al. 1987). Derivations appear as follows:³⁰

- (43)
- | | | |
|-------------------------|------------------------|------------|
| Gilbert | sees | George |
| ----- | | |
| NP:gilbert' | (S:see' y x\NP:x)/NP:y | NP:george' |
| -----> | | |
| S:see' george' x\NP:x | | |
| -----< | | |
| S:see' george' gilbert' | | |

(Where possible of course we suppress all the semantic detail.)

All the alternative derivations that the combinatory grammar permits yield equivalent semantic interpretations, representing the canonical function-argument relations that result from a purely applicative derivation. In contrast to combinatory derivations, such semantic representations therefore

²⁹Again there is a variable-free version of notation (a), using combinators in the semantics in place of λ , as in the following:

(i) $X/Y : f \quad Y/Z : g \Rightarrow X/Z : \mathbf{B}fg$

However the same objection applies.

³⁰For simplicity, we ignore type raising here.

preserve traditional notions of dominance and command, a point that has obviously desirable consequences if we wish to capture the generalisations concerning dependency that have been described in the GB framework in terms of relations of *c*-command. This point is important, for example, to the analysis of parasitic gaps sketched earlier, since parasitic gaps are known to obey an “anti-*c*-command” restriction.

The fact that such constraints can be regarded as holding over interpretations, as in the work of Bach 1979, 1980, Dowty 1982 and Chierchia 1988, as opposed to over surface structures, as in GB, is frequently unappreciated (see section 5 below), so it is worth dwelling on for a moment.

The interpretation of *Gilbert sees George* in the above derivation happens to directly reflect the dominance relations exhibited in a traditional surface structure for that sentence. This structure is stipulated in the lexical entry for the verb *sees*, 41. One might make all such interpretations, including those of control verbs, correspond to traditional surface structures in a similar way. But other arrangements could have been stipulated. For example, one might choose to have unordered “flat” argument structures, rather than the “curried” structures assumed here. A more attractive possibility, in view of the Montague work on binding and control mentioned above, and more recent work by Jacobson 1987, 1990, 1991, Pollard and Sag 1987, Hepple 1990, Szabolcsi 1992, and Dowty 1992, is to make dominance in such structures reflect the NP “obliqueness” hierarchy on grammatical relations, thus resembling the “argument structures” of Grimshaw 1990, and allowing the notion of *F*-command (Bach & Partee 1980) to be used in place of *c*-command. It follows that many of the classic theoretical issues of GB theory also find a very direct parallel in categorial terms in questions concerning the details of this representation. For example, if one is drawn to a PRO analysis of control, or wished to distinguish the subject as an “external” argument, in contrast to other “internal” arguments of the verb, it is here that the distinctions would appear. It is likely that many generalisations from GB and elsewhere concerning bounded constructions will transfer in this way, although it is to be hoped some of the degrees of freedom exploited in GB will not be required, given the very different treatments of long range dependencies that are available within CG.

By the very token that combinatory derivations *preserve* canonical rela-

tions of dominance and command, we must distinguish this level of semantic interpretation from the one implicated in the proposals of Geach 1972, Hausser 1984, Levin 1982, and Potts 1988. These authors use a very similar range of combinatory operations, notably including or entailing as theorems (generalised) functional composition, (lexical, polymorphic) type-lifting, and (in the case of Geach 1972, p.485) a coordination schema of the kind introduced in the previous section, in order to free the scope of quantifiers from traditional surface syntax, in order to capture the well-known ambiguity of sentences like the following:

(44) Every woman loves some man

On the simplest assumption that the verb is of type $e \rightarrow (e \rightarrow t)$, and the subject and object are corresponding (polymorphic) type-raised categories, the reading where the subject has wide scope is obtained by a purely applicative reading. The reading where the object has wide scope is obtained by composing subject and verb before applying the object to the result of the composition. In this their motivation for introducing composition is the combinatory relative of the λ -categorial grammars of Lewis, Montague, and Cresswell (see above). Indeed, we must sharply distinguish the *level* of semantic representation that is assumed in these two kinds of theory, as Lewis in fact suggested 1972, p.48, ascribing all these authors' operations to the level of logical form. Otherwise we must predict that those sentences which under the assumptions of the combinatory approach *require* function composition to yield an analysis (as opposed to merely allowing that alternative), such as right node raising, must yield only one of the two readings. (Which reading we get will depend upon the original assignment of categories). However, this prediction would be incorrect: both scopings are allowed for sentences like the following, adapted from Geach:

(45) Every girl likes, and every boy detests, some saxophonist.

That is not to say that the categorial analysis is without advantages. As Geach points out, we do *not* appear to obtain a third reading in which two instances of the existential each have wide scope over one of the universals, so that all the girls like one particular saxophonist, and all the boys detest one particular saxophonist, but the two saxophonists are not the same. This

result is to be expected if the entire substring *Every girl likes and every boy detests* is the syntactic and semantic constituent with respect to which the scope of the existential is defined. However, it remains the case that there is a many-to-one relation between semantic categories at this level and categories and/or rules at the level we have been considering up to now. The semantics itself and the nature of this relationship are a subject in their own right which it is not possible to do justice to here, but the reader is referred to important work by Partee & Rooth 1983 and Hendriks 1987 on the question. Much of this work has recently harkened back to axiomatic frameworks related to the Lambek calculus.

§5 CONCLUSIONS

Theories of the kind surveyed here have been applied with some success to a wide range of syntactic phenomena of the kinds touched on above in a number of languages, the latter including Dutch (Moortgat, Hoeksema, Hepple, and others cited above), Finnish (Karttunen, Jokinen, 1989), French (Desclés), Luiseno (Steele), Korean (Kang 1988), Spanish (Nishida 1), and Warlbiri (Bouma 1986).

Much criticism of theories in this area has been confounded with misconceptions, three of which are sufficiently widespread to require comment here. First, it is sometimes argued on the basis of the permutation completeness of van Benthem's calculus that categorial grammars overgeneralise. Of course, this is as absurd as claiming that *move- α* overgenerates. It is simply to mistake the true locus of the theoretical content. A more sophisticated version of this criticism claims that the restrictions on CG (for example, those in the Principles of Consistency and Inheritance) merely "simulate" constraints on movement (cf. von Stechow, 1990, p.473). I have pointed above to a certain broad resemblance of combinatory projection of directionality of government to the proposal of Kayne. It may of course be true that this resemblance amounts to nothing short of simulation of the empty category principle, projection principle, and the like. However, to prove that claim would require a careful comparison of the degrees of freedom exploited in this and the alternative theories, and of the generalisations that are captured, such as those concerning subject-object asymmetries, universal constraints on coordination, and others outlined above. Such careful comparison has not been

forthcoming.

A second criticism has arisen from the mistaken belief that phenomena that depend upon *c*-command, such as binding and control, cannot be captured in grammars with such flexible surface structures. While it is true that generalised categorial grammars tend towards structural completeness, and therefore allow objects, for example, to structurally command subjects at surface structure, we have seen that such non-standard structures are guaranteed to deliver interpretations that preserve traditional notions of dominance and command. It follows that all such regularities can be captured at the level of interpretation, as should be obvious from widespread similar proposals within Montague Grammar and Lexical-Functional Grammar. Of course, it can again be alleged (cf. von Stechow 1990, p.475) that such accounts merely simulate an S-structure-based account. However, the different treatment of long-distance dependency, and in particular the absence of *Wh*-trace from the relevant structures in the categorial account, means that a burden of proof still lies with the critics. Again, such proof has not been forthcoming.

A third vulgar error concerning these grammars is that they are disproportionately difficult to parse. It is certainly true that the inclusion of associative operations like composition means that for every analysis that is recognised by a traditional surface grammar, there are in general several semantically equivalent but derivationally distinct categorial analyses, a phenomenon which is misleadingly referred to as “spurious” ambiguity. The “forest” of alternatives that must be searched to ensure that all possible readings of a sentence are derived is potentially very large, because the grammar is highly non-deterministic. Serious though this problem is for practical computational applications, it is a mistake to think that it is peculiar to categorial grammar. *Any* theory that captures a comparable range of constructions must necessarily encounter exactly the same degree of structural ambiguity. Far from being “spurious”, it is a fact of competence grammar.

Finally, it might be suggested that combinators are notationally cumbersome by comparison with the λ -calculus, and hence intrinsically unlikely to be primitive operations of cognition. However, from a psychological and evolutionary point of view, they seem very good candidates, for they can plausibly be argued to be individually useful for cognition in general. For

example, since actions can be regarded as functions from states to states, then the achievement of a compound action, even one as simple as reaching one's arm around an obstacle, can be regarded as requiring the composition of more primitive actions. One might expect it to be simpler for evolution to give rise to the specific capability of composing functions than to a completely general-purpose abstraction operator like λ . One may therefore speculate that the concept-formation mechanism has taken a combinatory form because it has evolved in a piecemeal fashion, out of elements that were selected for more restricted functions, and that this preproperty is inherited by the linguistic system. If so, the combinators may prove to be not only the "building blocks" of mathematical logic as Schönfinkel 1924 claimed, and of natural language as is claimed here, but of even more fundamental cognitive faculties.

REFERENCES

- Ades, Anthony E. and Mark J. Steedman: 1982, 'On the Order of Words', *Linguistics and Philosophy*, **4**, 517-558.
- Ajdukiewicz, Kazimierz: 1935, 'Die syntaktische Konnexität', *Studia Philosophica*, **1**, 1-27. English translation in Storrs McCall (ed), *Polish Logic 1920-1939*, Oxford University Press, pp. 207-231.
- Bach, Emmon: 1979, 'Control in Montague Grammar', *Linguistic Inquiry*, **10**, 513-531.
- Bach, Emmon: 1980, 'In Defense of Passive', *Linguistics and Philosophy*, **3**, 297-341.
- Bach, Emmon, and Barbara Partee: 1980, 'Anaphora and Semantic Structure', in *Papers from the Parasession on Pronouns and Anaphora, Tenth Meeting of the Chicago Linguistics Society*, CLS, Chicago.
- Bar-Hillel, Yehoshua: 1953, 'A Quasi-arithmetical Notation for Syntactic Description', *Language*, **29**, 47-58.
- Bar-Hillel, Yehoshua: 1960, 'Some Linguistic Obstacles to Machine Translation', in Yehoshua Bar-Hillel, *Language and Information*, Addison-Wesley, New York. 75-86.
- Bar-Hillel, Yehoshua, Chaim Gaifman, and Eliyahu Shamir: 1960, 'On Categorical and Phrase-Structure Grammars, in Yehoshua Bar-Hillel, *Language and Information*, Addison-Wesley, New York. 99-116.
- Barry, Guy: 1991, *Derivation and Structure in Categorical Grammar*, unpublished Ph.D dissertation, University of Edinburgh.
- Bartsch Renate, and Theo Venneman: 1972, *Semantic Structures*, Athenäum, Frankfurt.
- van Benthem, Johan: 1986, *Essays in Logical Semantics*, Reidel, Dordrecht.
- van Benthem, Johan: 1988, 'The Lambek Calculus', in Oehrle et al. 1988, 35-68.
- van Benthem, Johan: 1991, *Language in Action*, North Holland, Amsterdam.
- Bouma, Gosse: 1986, 'Warlbiri Wildness', ms. Rijkuniversiteit Groningen.
- Bouma, Gosse: 1987, 'A Unification-based Analysis of Unbounded Dependencies in Categorical Grammar', in J. Groenendijk, M. Stokhof, and F. Veltman, (eds.), *Proceedings of the 6th Amsterdam Colloquium*, Foris, Dordrecht.

- Bresnan, Joan (ed.): 1982, *The mental Representation of Grammatical Relations*, MIT Press, Cambridge, MA.
- Buszkowski, Wojciech: 1982, *Lambek's Categorical Grammars*, dissertation, Adam Mickiewicz University, Poznań.
- Buszkowski, Wojciech: 1988, 'Generative Power of Categorical Grammars', in Oehrle et al, 1988, 69-94.
- Buszkowski, Wojciech, Witold Marciszewski, and Johan van Benthem (eds): 1988, *Categorical Grammar*, Amsterdam, John Benjamins.
- Carpenter, Robert: 1989, *Phrase Meaning and Categorical Grammar*, unpublished Ph.D dissertation, University of Edinburgh.
- Chierchia, Gennaro: 1985, 'Formal Semantics and the Grammar of Predication', *Linguistic Inquiry*, **16**, 417-443.
- Chierchia, Gennaro: 1988, 'Aspects of a Categorical Theory of Binding', in Oehrle et al, 1988, 153-98.
- Chomsky, Noam: 1963, 'Formal Properties of Grammars', in R. Duncan Luce et al. (eds.), *Handbook of Mathematical Psychology*, vol. 2, Wiley, New York.
- Chomsky, Noam: 1986, *Barriers*, MIT Press, Cambridge MA.
- Cormack, Annabel: 1986, 'VP Anaphora: Variables and Scope',
- Cresswell, Max: 1973, *Logics and Languages*, Methuen.
- Curry, Haskell, and Robert Feys: 1958, *Combinatory Logic: Vol I*, North Holland, Amsterdam.
- Curry, Haskell: 1961, 'Some Logical Aspects of Grammatical Structure', in R. Jakobson (ed.), *Structure of Language and its Mathematical Aspects, Proceedings of the Symposium in Applied Mathematics, XII*, American Mathematical Society, Providence, Rhode Island. 56-68.
- Dahl, Östen: 1977, *Logic, Pragmatics, and Grammar*, Studentlitteratur, Lund.
- Desclés, Jean-Pierre, Zlatka Guentcheva, and Sebastian Shaumyan: 1986, 'Theoretical Analysis of Reflexivisation in the Framework of Applicative Grammar', *Linguisticæ Investigationes*, **10**, 1-65.
- Dowty, David: 1982, 'Grammatical Relations and Montague Grammar', in Pauline Jacobson and Geoffrey K. Pullum (eds.), *The Nature of Syntactic Representation*, Reidel, Dordrecht. 79-130.
- Dowty, David: 1988, 'Type-raising, Functional Composition, and Non-constituent Coordination', in Oehrle et al, 1988, 153-98.
- Dowty, David: 1992, "Variable-free" Syntax, Variable-binding Syntax, the

- Natural Deduction Lambek Calculus, and the Crossover Constraint', in *Proceedings of the 1992 West Coast Conference on Formal Linguistics*, SLA Stanford CA, to appear.
- Flynn, Michael: 1983, 'A Categorical Theory of Structure Building', in Gerald Gazdar, E. Klein and G.K. Pullum, (eds.), *Order, Concord, and Constituency*, Foris, Dordrecht. 139-174.
- Friedman, Joyce, D. Dai, and W. Wang: 1986, 'The Weak Generative Capacity of Parenthesis-free Categorical Grammars', *Proceedings of COLING 1986*. 199-201.
- Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum, and Ivan A. Sag: 1985, *Generalised Phrase Structure Grammar*, Blackwell, Oxford.
- Geach, Peter T: 1972, 'A Program for Syntax', in Donald Davidson and Gilbert Harman, *Semantics of Natural Language*, Reidel, Dordrecht. 483-497 (reprinted in Buszkowski et al. 1988).
- Gentzen, G: 1934, 'Untersuchungen über das Logische Schliessen', *Mathematische Zeitschrift*, **39**, 176-210, 405-431.
- Grimshaw, Jane: 1990, *Argument Structure*, MIT Press, Boston MA.
- Hausser, Roland: 1984 *Surface Compositional Grammar* Wilhelm Fink Verlag, München.
- Hepple, Mark: 1990, *The Grammar and Processing of Order and Dependency: a Categorical Approach*, Ph.D dissertation, University of Edinburgh. Memo, Centre for Cognitive Science, University of Edinburgh.
- Hendriks, Herman: 1987, 'Type-change in Semantics' in Ewan Klein and J. van Benthem (eds.), *Categories, Polymorphism, and Unification*, Centre for Cognitive Science, University of Edinburgh.
- Hiz, Henry: 1967, 'Grammar Logicism', *The Monist*, **51**, 110-127 (reprinted in Buszkowski et al. 1988).
- Hoeksema, Jack: 1985, *Categorical Morphology* PhD dissertation, Rijksuniversiteit, Groningen, publ. by Garland, New York.
- Hoeksema, Jack: 1991, 'Complex Predicates and Liberation in Dutch and English', *Linguistics and Philosophy*, **14**, 661-710.
- Huck, Geoffrey: 1988, 'Phrasal Verbs and the Categories of Postponement', in Oehrle et al, 1988, 249-264.
- Hudson, Richard: 1984, *Word Grammar*, Blackwell, Oxford.
- Jacobson, Pauline: 1987, 'Phrase Structure, Grammatical Relations and Discontinuous Constituents', in Geoffrey Huck & Almerido Ojeda, (eds.) *Syntax and Semantics 20: Discontinuous Constituency*, Academic Press,

- New York, 27-69.
- Jacobson, Pauline: 1990, 'Raising as Function Composition', *Linguistics & Philosophy*, **13**, 423-476.
- Jacobson, Pauline: 1991, 'Bach-Peters sentences in a Variable-free semantics', *Proceedings of the Eighth Amsterdam Colloquium*, to appear.
- Jokinen, Kristiina: 1988, 'Lexicon and Morphosyntax', in Jussi Niemi (ed.) *Papers for the 11th Scandinavian Conference of Linguistics*, University of Joensuu, Joensuu.
- Joshi, Aravind, K. Vijay-Shanker, and David Weir: 1991, 'The Convergence of Mildly Context-sensitive Formalisms', in Peter Sells, Stuart Shieber and Tom Wasow, (eds.), *Processing of Linguistic Structure*, MIT Press, Cambridge MA. 32-81.
- Kang, Beom-Mo: 1988, *Functional Inheritance, Anaphora, and Semantic Interpretation in a Generalised Categorical Grammar*, Ph.D Thesis, Brown University, Providence RI.
- Kaplan, David: 1975, 'How to Russell a Frege-Church', *Journal of Philosophy*, **72**, 716-729.
- Karttunen, Lauri: 1989, 'Radical Lexicalism', in Mark Baltin and A. Kroch (eds.), *Alternative Conceptions of Phrase Structure*, Chicago University Press. 43-65.
- Kayne, Richard: 1984, *Connectedness and Binary Branching*, Foris, Dordrecht.
- Keenan, Ed: 1971, 'Quantifier Structures in English', *Foundations of Language*, **7**, 255-284.
- Keenan, Ed and Leonard Faltz: 1985, *Boolean Semantics for Natural Language*, Reidel, Dordrecht.
- Kleene, S: 1952, *Introduction to Metamathematics*, van Nostrand, New York.
- Lambek, Joachim: 1958, 'The Mathematics of Sentence Structure', *American Mathematical Monthly*, **65**, 154-170.
- Lambek, Joachim: 1961, 'On the Calculus of Syntactic Types', in R. Jakobson (ed.), *Structure of Language and its Mathematical Aspects, Proceedings of the Symposium in Applied Mathematics, XII*, American Mathematical Society, Providence, Rhode Island. 166-178.
- Lambek, Joachim: 1990, 'On a Connection between Algebra, Logic, and Linguistics' ms. McGill University, Montreal.
- Lambek, Joachim: 1990, 'Review of Moortgat 1988b', ms. McGill University, Montreal.

- Levin, Harold: 1982, *Categorial Grammar and the Logical Form of Quantification*, Bibliopolis, Milan.
- Lewis, David: 1972, 'General Semantics' in Donald Davidson and Gilbert Harman, *Semantics of Natural Language*, Reidel, Dordrecht, p.169-218.
- Lyons, John: 1968, *Theoretical Linguistics*, Cambridge University Press.
- Montague, Richard: 1970, 'Universal Grammar' *Theoria*, **36**, 373-398.
- Montague, Richard: 1973, 'The Proper Treatment of Quantification', in Jaako Hintikka, J. Moravcsik and P. Suppes (eds.), *Approaches to Natural Language*, Reidel, Dordrecht. 221-242.
- Moortgat, Michael: 1988a, 'Mixed Composition and Discontinuous Dependencies', in Oehrle et al, 1988, 319-348.
- Moortgat, Michael: 1988b, *Categorial Investigations*, Ph.D dissertation, University of Amsterdam, publ. Foris, Dordrecht, 1989.
- Morrill, Glyn: 1988, *Extraction and Coordination in Phrase Structure Grammar and Categorial Grammar*, unpublished Ph.D dissertation, University of Edinburgh.
- Nishida, Chiyo: 1990, 'A non-transformational analysis of clitic climbing in Spanish', in A.L. Halpern (ed.), *Proceedings of the Ninth West Coast Conference on Formal Linguistics*, Stanford Linguistics Association, Stanford. 395-409.
- Oehrle, Richard T: 1987, 'Boolean Properties in the Analysis of Gapping', in Geoffrey Huck and A. Ojeda (eds), *Syntax and Semantics 20: Discontinuous Constituency*, Academic, New York. 203-240.
- Oehrle, Richard T: 1988a, 'Multidimensional Compositional Functions as a Basis for Grammatical Analysis', in Oehrle et al, 1988. 349-390.
- Oehrle, Richard T: 1990, 'Categorial frameworks, coordination, and extraction', in A.L. Halpern (ed.), *Proceedings of the Ninth West Coast Conference on Formal Linguistics*, Stanford Linguistics Association, Stanford. 411-425.
- Oehrle, Richard T, Emmon Bach and Deirdre Wheeler, (eds), *Categorial Grammars and Natural Language Structures*, papers from the Conference on Categorial Grammar, May 1985, Tucson, Reidel, Dordrecht.
- O'Grady, William: 1991, *Categories and Case*, John Benjamins, Amsterdam.
- Partee, Barbara, and Mats Rooth: 1983, 'Generalised Conjunction and Type Ambiguity, in Rainer Bäuerle, C. Schwarze and A. von Stechow (eds.), *Meaning, Use and Interpretation of Language*, de Gruyter, Berlin. 361-383.

- Pollard, Carl and Ivan Sag: 1987, *Information-based Syntax and Semantics*, CSLI/Chicago University Press, Chicago.
- Potts, Timothy: 1988, 'Fregean Grammar: a Formal Outline', in Buszkowski et al. 1988, 221-244.
- Schabes, Yves, Anne Abeillé, and Aravind Joshi: 1988, 'Parsing Strategies with "Lexicalised" Grammars: Application to Tree Adjoining Grammars', *Proceedings of the 12th International Conference on Computational Linguistics*, Coling, Budapest.
- Schönfinkel, Moses: 1924, 'On the Building Blocks of Mathematical Logic', in Jean van Heijenoort (ed.), *From Frege to Gödel*, Harvard University Press, Cambridge MA. 355-366.
- Shaumyan, Sebastian: 1977, *Applicational Grammar as a Semantic Theory of Natural Language*, Edinburgh University Press.
- Shieber, Stuart: 1986, *An Introduction to Unification-based Approaches to Grammar*, CSLI/Chicago University Press, Chicago.
- Smullyan, Raymond: 1985, *To Mock a Mockingbird*, Knopf, New York.
- von Stechow, Arnim: 1974, ' $\epsilon - \lambda$ -kontextfreie Sprachen: ein Beitrag zu einer natürlich formalen Semantik', *Linguistische Berichte*, **34**, 1-33.
- von Stechow, Arnim: 1990, 'Categorial Grammar and Linguistic Theory', *Studies in Language*, **14**, 433-478.
- Steele, Susan: 1990, *Agreement and Anti-agreement*, Reidel, Dordrecht.
- Steedman, Mark: 1991a, 'Structure and Intonation', *Language*, **67**, 262-296.
- Szabolcsi, Anna: 1983, 'ECP in Categorial Grammar', ms, Max Planck Institute, Nijmegen.
- Szabolcsi, Anna: 1987, 'Bound Variables in Syntax: Are there any?' in R. Bartsch, J. van Benthem, and P. van Emde Boas (eds.), *Semantics and Contextual Expression*, 294-318, Foris, Dordrecht.
- Szabolcsi, Anna: 1987, 'On Combinatory Categorial Grammar', *Proceedings of the Symposium on Logic and Language, Debrecen*, Akadémiai Kiadó, Budapest, 151-162.
- Szabolcsi, Anna, and Franz Zwarts: 1990. Semantic Properties of Composed Functions and the Distribution of *Wh*-phrases. Proceedings of the Seventh Amsterdam Colloquium, ed. by Stokhof and Torenvliet. 529-555. Amsterdam: ILLI.
- Szabolcsi, Anna: 1991, 'Combinatory Categorial Grammar and Projection from the Lexicon' in Ivan Sag & Anna Szabolcsi, (eds.), *Lexical Matters*, CSLI, Stanford CA.

- Uszkoreit, Hans: 1986, 'Categorial Unification Grammars', *Proceedings of the 11th International Conference on Computational Linguistics*, Bonn, August, 1986, 187-194.
- Weir, David: 1988, *Characterising Mildly Context-sensitive Grammar Formalisms*, Ph.D dissertation, CIS-88-74, University of Pennsylvania.
- Weir, David, and Aravind Joshi: 1988, 'Combinatory Categorial Grammars: Generative Power and Relation to Linear CF Rewriting Systems', *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, June 1988, 278-285.
- Wittenburg, Kent: 1986, *Natural Language Parsing with a Combinatory Categorial Grammar in a Graph-unification-based Formalism*, unpublished PhD thesis, University of Texas at Austin.
- Wood, M. 1988, *A Categorial Syntax for Coordinate Constructions*, Ph. D dissertation, University College, London.
- Uszkoreit, Hans: 1986, 'Categorial Unification Grammar', *Proceedings of the 11th International Conference on Computational Linguistics*, Bonn August 1986, 187-194.
- Zeevat, Henk, Ewan Klein, and Jo Calder: 1987, 'An Introduction to Unification Categorial Grammar', in N. Haddock et al. (eds.), *Edinburgh Working Papers in Cognitive Science, 1: Categorial Grammar, Unification Grammar, and Parsing*.
- Zielonka, W: 1981, 'Axiomatisability of Ajdukiewicz-Lambek Calculus by Means of Cancellation Schemes', *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, **27**, 215-224.
- Zwarts, Franz: 1986, *Categoriale Grammatica en Algebraïsche Semantik*, Ph.D dissertation, Rijkuniversiteit, Groningen.